



Integer programming techniques for educational timetabling

Fonseca, George H.G.; Santos, Haroldo G.; Carrano, Eduardo G.; Stidsen, Thomas Jacob Riis

Published in:
European Journal of Operational Research

Link to article, DOI:
[10.1016/j.ejor.2017.03.020](https://doi.org/10.1016/j.ejor.2017.03.020)

Publication date:
2017

Document Version
Peer reviewed version

[Link back to DTU Orbit](#)

Citation (APA):
Fonseca, G. H. G., Santos, H. G., Carrano, E. G., & Stidsen, T. J. R. (2017). Integer programming techniques for educational timetabling. *European Journal of Operational Research*, 262(1), 28-39.
<https://doi.org/10.1016/j.ejor.2017.03.020>

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Integer Programming Techniques for Educational Timetabling

George H. G. Fonseca^{a,b,c}, Haroldo G. Santos^d, Eduardo G. Carrano^{e,a},
Thomas J.R. Stidsen^c

^a*Graduate Program in Electrical Engineering
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brazil*

^b*Department of Computing and Information Systems
Universidade Federal de Ouro Preto
St. Diogo de Vasconcelos 328, 35400-000
Ouro Preto, MG, Brazil*

^c*Department of Management Engineering
Technical University of Denmark
Building 426, Produktionstorvet, DK-2800
Kgs. Lyngby, Denmark*

^d*Department of Computing
Universidade Federal de Ouro Preto
St. Diogo de Vasconcelos 328, 35400-000
Ouro Preto, MG, Brazil*

^e*Department of Electrical Engineering
Universidade Federal de Minas Gerais
Av. Antônio Carlos 6627, 31270-901
Belo Horizonte, MG, Brazil*

Abstract

Educational timetabling problems require the assignment of times and resources to events, while sets of required and desirable constraints must be considered. The XHSTT format was adopted in this work because it models the main features of educational timetabling and it is the most used format in recent studies in the field. This work presents new cuts and reformulations for the existing integer programming model for XHSTT. The proposed

Email addresses: george@decsi.ufop.br (George H. G. Fonseca),
haroldo@iceb.ufop.br (Haroldo G. Santos), egcarrano@ufmg.br (Eduardo G.
Carrano), thst@dtu.dk (Thomas J.R. Stidsen)

cuts improved hugely the linear relaxation of the formulation, leading to an average gap reduction of 32%. Applied to XHSTT-2014 instance set, the alternative formulation provided four new best known lower bounds and, used in a matheuristic framework, improved eleven best known solutions. The computational experiments also show that the resulting integer programming models from the proposed formulation are more effectively solved for most of the instances.

Keywords: Timetabling, Integer Programming, Formulation

1. Introduction

Educational timetabling problems consist in assigning times and resources to events of educational institutions respecting several hard and soft constraints. Sometimes this problem can be solved manually by the institution manager. However, this task becomes virtually impossible when there are hundreds of events, resources, and constraints to be handled. Moreover, the use of optimization techniques to generate timetables leads to significantly better schedules, allowing the institution to use more efficiently its resources, to improve the satisfaction of the staff, and even to increase the performance of the student. Beyond its practical importance, this problem is \mathcal{NP} -Hard [1], which makes it of interest for Operations Research and Artificial Intelligence communities.

Studies on timetabling optimization have begun in 1963 by Gotlieb [2] and, since such a work, several approaches have been developed to tackle this problem. Most of these approaches are based on metaheuristics, such as Simulated Annealing [3, 4], Greedy Randomized Adaptive Search Procedure [5], Tabu Search [6, 7], Variable Neighborhood Search [8, 9], Iterated Local Search [10, 11] and Evolutionary Strategies [12, 13]. More recently, model-based heuristics (so-called Matheuristics) have achieved remarkable results for this problem class [14, 15].

Although metaheuristics frequently achieve reasonable solutions in short time, they do not ensure optimality. Moreover, they cannot even provide optimality gaps. In this sense, exact techniques, such as Constraint Programming and Mathematical Programming, take an important role. They can provide lower bounds to evaluate more precisely the quality of a solution and even reach good or optimal solutions. However, to achieve good/optimal solutions through exact methods may be possible only when the timetabling

problem is small or when advanced techniques of formulation, such as Column Generation [16], Cuts [17, 18], and Decomposition [19] are applied.

This work considered the eXtended Markup Language for High School Timetabling (XHSTT) format [20] for modelling the problem. This format was chosen due to its capacity of covering several timetabling features and due to its relevance in recent literature, being adopted by the last international timetabling competition [21]. Although it was initially proposed for High School Timetabling, it can be also applicable to universities and other types of educational institutions, such as shown in [22]. Indeed, several XHSTT problem sets found in the literature, including the one considered in this paper, contain university timetabling, student sectioning and other instances that are not necessarily from high school institutions. Finally, several instances are available in this format and the research community keeps track of the best known bounds for them.

Kristiansen *et al.* [23] proposed the first Integer Programming formulation for the XHSTT format. However, to the best of our knowledge, not much effort has been done to strengthen Kristiansen’s formulation or to develop alternative formulations. In this sense, the objective of this work is to propose new cuts and reformulations to the existing XHSTT Integer Programming formulation.

The remainder of the paper is organized as follows. The XHSTT timetabling problem as well as its complete formulation are described in Section 2. The alternative formulation and the implemented cuts are presented in Section 3. Computational experiments and results achieved are reported in Section 4. Finally, some concluding remarks are drawn in Section 5.

2. Complete Formulation

Kristiansen’s formulation [23] was designed to describe precisely and to handle any instance in the XHSTT format. This formulation is presented in this section and denoted as \mathcal{F}_1 throughout the paper. The formulation takes as input sets:

An event $e \in \mathcal{E}$ has a duration $D_e \in \mathbb{N}$ and a demand for a set of resources (event resources), denoted as $er \in \mathcal{ER}_e$. Furthermore, a resource demanded for the event er can undertake a role $role_{er}$, which is used to link the resource to certain constraints. A resource $r \in \mathcal{R}$ can be preassigned to fulfil the demand $er \in \mathcal{ER}_e$. Parameter $PA_{er} \in \{1, 0\}$ take value 1 if event resource er has a preassigned resource, and 0 otherwise, while parameter

\mathcal{T}	Times
\mathcal{TG}	Time Groups
\mathcal{R}	Resources
\mathcal{RG}	Resource Groups
\mathcal{E}	Events
\mathcal{EG}	Event Groups
\mathcal{C}	Constraints

$PR_{er,r} \in \{1, 0\}$ take value 1 if resource r is preassigned to event resource er , and 0 otherwise. A sub-event se is defined as a fragment of a specific event $e \in \mathcal{E}$. Each sub-event has a duration $D_{se} \leq D_e$ and it inherits exactly the same resource requirements of the source event.

Let \mathcal{SE} be the entire set of sub-events of a XHSTT instance and let $se \in \mathcal{SE}_e$ be a set of the sub-events generated from an event e . The total duration of all sub-events generated from event e in a solution must be exactly D_e . A set of all possible sub-events with different durations is created, such that all combinations of sub-events for a given event can be handled. For example, if an event has duration 4, the set of sub-events for this event has the respective lengths: 1, 1, 1, 1, 2, 2, 3 and 4. Thereby the set of possible sub-events for an event $e \in \mathcal{E}$ with duration D_e has $\sum_{i=1}^{D_e} \lfloor \frac{D_e}{i} \rfloor$ elements.

The times \mathcal{T} are organized in chronological time; thus, p_t denotes the index number of time $t \in \mathcal{T}$. A time group $tg \in \mathcal{TG}$ defines a set of times, in such a way $t \in \mathcal{T}_{tg}$ denotes the times belonging to time group tg . Additionally

$$\mathcal{T}_{se,t}^{start} = \{t' \in \mathcal{T} \setminus \{t_D\} \mid p_t - D_{se} + 1 \leq p_{t'} \leq p_t\} \quad (1)$$

is pre-processed to denote the set of times that a sub-event $se \in \mathcal{SE}$ occurs assuming that it is assigned start time $t \in \mathcal{T}$.

Each constraint $c \in \mathcal{C}$ is of a specific type and it applies to certain events, resources or event groups. Notations $e \in \mathcal{E}_c$, $r \in \mathcal{R}_c$ and $eg \in \mathcal{EG}_c$ represent, respectively, the events, resources or event groups that a constraint applies to.

The set of resources and times are both extended with dummy-indices, denoted dummy-resource r_D and dummy-time t_D , respectively. They are necessary to handle with the unusual case of an optimal solution in which one or more events do not have resources or start times assigned.

2.1. Variables

The main decision variables of this formulation are $x_{se,t,er,r}$ binary variables:

$$x_{se,t,er,r} = \begin{cases} 1 & \text{if sub-event } se \text{ is assigned to start time } t \text{ and} \\ & \text{resource } r \text{ is assigned to event resource } er \\ 0 & \text{otherwise} \end{cases}$$

additionally, the following support variables are used:

$$\begin{aligned} y_{se,t} &= 1 \text{ if sub-event } se \text{ is assigned to start time } t; 0 \text{ otherwise.} \\ w_{se,er,r} &= 1 \text{ if sub-event } se \text{ is assigned to resource } r \text{ for event resource } er; 0 \text{ otherwise.} \\ u_{se} &= 1 \text{ if sub-event } se \text{ is active; 0 otherwise.} \\ v_{t,r} &= \text{number of times in which resource } r \text{ is used at time } t. \\ q_{r,t} &= 1 \text{ if resource } r \text{ is busy at time } t; 0 \text{ otherwise.} \\ p_{r,tg} &= 1 \text{ if resource } r \text{ is busy at at least one time of time group } tg; 0 \text{ otherwise.} \\ o_{e,t} &= 1 \text{ if at least one sub-event of event } e \text{ is assigned to time } t; 0 \text{ otherwise.} \\ l_{eg,t} &= 1 \text{ if at least one event of event group } eg \text{ is assigned to time } t; 0 \text{ otherwise.} \\ k_{eg,r} &= 1 \text{ if resource } r \text{ is assigned to at least one event in event group } eg; 0 \text{ otherwise.} \\ h_{r,tg,t} &= 1 \text{ if resource } r \text{ has an idle time in time } t \text{ in time group } tg; 0 \text{ otherwise.} \\ h_e^{first} &= \text{ordinal number of the first time assigned to any sub-event of event } e. \\ h_e^{last} &= \text{ordinal number of the latest time assigned to any sub-event of event } e. \end{aligned}$$

Each constraint $c \in \mathcal{C}$ has a set of points-of-application which, in turn, might be any XHSTT entity depending on the constraint that it is related to. To simplify the notation, in some equations points-of-application will be denoted as $p \in \mathcal{P}_c$ regardless of the entity. Each point-of-application is associated with a set of deviations, indexed by $d \in \mathcal{D}_p$, and each set of deviations has an associated non-negative cost. Therefore the slack variables

$$s_{c,p,d} = \text{value of deviation } d \text{ of point-of-application } p \text{ in constraint } c.$$

are used to calculate the penalties for each XHSTT constraint.

2.2. Constraints

In addition to all constraints described in the XHSTT specification, some basic constraints are required to ensure the consistency of the model. They are shown in equations (2) to (14).

$$\sum_{t \in \mathcal{T}} \sum_{r \in \mathcal{R}_{er}} x_{se,t,er,r} = 1 \quad \begin{matrix} \forall se \in \mathcal{SE} \\ \forall er \in \mathcal{ER}_{se} \end{matrix} \quad (2)$$

$$\sum_{er \in \mathcal{ER}_{se}} \sum_{r \in \mathcal{R}_{er}} x_{se,t,er,r} = |\mathcal{ER}_{se}| \times y_{se,t} \quad \begin{matrix} \forall se \in \mathcal{SE} \\ \forall t \in \mathcal{T} \end{matrix} \quad (3)$$

$$\sum_{se \in \mathcal{SE}} \sum_{er \in \mathcal{ER}_{se}} \sum_{t' \in \mathcal{T}_{se,t}^{start}} x_{se,t',er,r} = v_{t,r} \quad \begin{matrix} \forall t \in \mathcal{T} \setminus \{t_D\} \\ \forall r \in \mathcal{R} \end{matrix} \quad (4)$$

$$\sum_{t \in \mathcal{T}} x_{se,t,er,r} = w_{se,er,r} \quad \begin{matrix} \forall se \in \mathcal{SE} \\ \forall er \in \mathcal{ER}_{se} \\ \forall r \in \mathcal{R} \end{matrix} \quad (5)$$

$$y_{se,t} = 0 \quad \begin{matrix} \forall se \in \mathcal{SE} \\ \forall t \in \mathcal{T} \setminus \{t_D\} : p_t + D_{se} - 1 > |\mathcal{T}| \end{matrix} \quad (6)$$

$$\sum_{r \in er \setminus \{r_D\}} w_{se,er,r} \leq u_{se} \quad \begin{matrix} \forall se \in \mathcal{SE} \\ \forall er \in \mathcal{ER}_{se} : PA_{er} = 0 \end{matrix} \quad (7)$$

$$\sum_{t \in \mathcal{T} \setminus \{t_D\}} y_{se,t} \leq u_{se} \quad \forall se \in \mathcal{SE} \quad (8)$$

$$\sum_{t \in \mathcal{T} \setminus \{t_D\}} y_{se,t} + \sum_{er \in \mathcal{ER}_{se} : PA_{er} = 0} \sum_{r \in \mathcal{R}_{er} \setminus \{r_D\}} w_{se,er,r} \geq u_{se} \quad \forall se \in \mathcal{SE} \quad (9)$$

$$\sum_{se \in \mathcal{SE}_e} u_{se} \times D_{se} = D_e \quad \forall e \in \mathcal{E} \quad (10)$$

$$|\mathcal{SE}| \times q_{r,t} \geq v_{t,r} \quad \begin{matrix} \forall r \in \mathcal{R} \\ \forall t \in \mathcal{T} \setminus \{t_D\} \end{matrix} \quad (11)$$

$$q_{r,t} \leq v_{t,r} \quad \begin{matrix} \forall r \in \mathcal{R} \\ \forall t \in \mathcal{T} \setminus \{t_D\} \end{matrix} \quad (12)$$

$$p_{r,tg} \geq q_{r,t} \quad \begin{matrix} \forall r \in \mathcal{R} \\ \forall tg \in \mathcal{TG} \\ \forall t \in \mathcal{T}_{tg} \end{matrix} \quad (13)$$

$$p_{r,tg} \leq \sum_{t \in \mathcal{T}_{tg}} q_{r,t} \quad \begin{matrix} \forall r \in \mathcal{R} \\ \forall tg \in \mathcal{TG} \end{matrix} \quad (14)$$

Constraint set (2) is necessary to make sure that only one starting time is assigned to any sub-event. Constraint set (3) makes the link between variables $x_{se,t,er,r}$ and $y_{se,t}$ whereas the number of event resources demanded for a sub-event se is denoted by $|\mathcal{ER}_{se}|$. Constraint set (4) links variables $x_{se,t,er,r}$ to variables $v_{t,r}$. The link between variables $x_{se,t,er,r}$ and $w_{se,er,r}$ is

done through constraint set (5). Constraint set (6) ensures that a starting time $t \in \mathcal{T}$ is not assigned to a sub-event $se \in \mathcal{SE}$ if there is not enough contiguous times after t to accommodate the duration of se .

Although all possible sub-events for an event are created, only a subset of them should be active in the final solution. Recall that a sub-event is active if a starting time or a resource is assigned to it. Constraint sets (7), (8) and (9) are imposed to ensure the right activation of the sub-events. Constraint set (10) ensures that the sum of the durations of the sub-events of a given event is equal to the duration of the source event.

A resource is busy at some time if it attends to at least one event on that time, and it is busy at some time group if it is busy at one or more times within the times of that time group. This relation is represented by variables $q_{r,t}$ and $p_{r,tg}$, which are set by constraint sets (11), (12), (13), and (14).

In sequence, each XHSTT specific constraint type is formulated. Let set $\mathcal{C}^i \subseteq \mathcal{C}$ denote all constraints of a certain type, as follows:

\mathcal{C}^1 – **Assign Resource** : An assign resource constraint penalizes a solution when no resource is assigned to supply a demand of an event resource. Specifically, the deviation at one point-of-application is the sum of the durations of the sub-events of the respective event in which a resource is not assigned. The deviation $s_{c,er}^1$ at each point-of-application of this constraint is calculated through the following equations:

$$D_e - \sum_{se \in \mathcal{SE}_e} \sum_{r \in \mathcal{R}_{er} \setminus \{r_D\}} D_{se} \times w_{se,er,r} = s_{c,er}^1 \quad \begin{array}{l} \forall c \in \mathcal{C}^1 \\ \forall e \in \mathcal{E}_c \\ \forall er \in \mathcal{ER}_e : role_{er} = role_c \end{array} \quad (15)$$

\mathcal{C}^2 – **Assign Time** : The assign time constraint penalizes sub-events in which times are not assigned. The deviation $s_{c,e}^2$ at one point-of-application is the total duration of those sub-events derived from the specific event in which a time is not assigned.

$$D_e - \sum_{t \in \mathcal{T} \setminus \{t_D\}} \sum_{se \in \mathcal{SE}_e} D_{se} \times y_{se,t} = s_{c,e}^2 \quad \begin{array}{l} \forall c \in \mathcal{C}^2 \\ \forall e \in \mathcal{E}_c \end{array} \quad (16)$$

\mathcal{C}^3 – **Split Events** : A split event constraint defines limits to the number of sub-events that can be derived from a given event and to their durations. Let the parameters $\underline{B}_c^{amt} \in \mathbb{N}$ and $\overline{B}_c^{amt} \in \mathbb{N}$ be respectively, the minimum and the maximum number of sub-events in which a given

event can be split, and $\underline{B}_c^{dur} \in \mathbb{N}$ and \overline{B}_c^{dur} be the minimum and maximum durations of such sub-events.

The value of the deviation at each point-of-application (each event) of this constraint is given by the number of sub-events of the source event whose duration is lower than \underline{B}_c^{dur} or greater than \overline{B}_c^{dur} ($s_{c,e}^{3a}$) and the number of sub-events below \underline{B}_c^{amt} , or above \overline{B}_c^{amt} ($s_{c,e}^{3b}$). The following constraint sets are imposed:

$$\sum_{\substack{se \in \mathcal{SE}_e : \\ \underline{B}_c^{dur} > D_{se} \vee \overline{B}_c^{dur} < D_{se}}} u_{se} = s_{c,e}^{3a} \quad \forall c \in \mathcal{C}^3, \forall e \in \mathcal{E}_c \quad (17)$$

$$\underline{B}_c^{amt} - \sum_{se \in \mathcal{SE}_e} u_{se} \leq s_{c,e}^{3b} \quad \forall c \in \mathcal{C}^3, \forall e \in \mathcal{E}_c \quad (18)$$

$$\sum_{se \in \mathcal{SE}_e} u_{se} - \overline{B}_c^{amt} \leq s_{c,e}^{3b} \quad \forall c \in \mathcal{C}^3, \forall e \in \mathcal{E}_c \quad (19)$$

The full deviation for constraint $c \in \mathcal{C}^3$ is given by $s_{c,e}^{3a} + s_{c,e}^{3b}$.

\mathcal{C}^4 – **Distribute Split Events** : Distribute split event constraints impose limits on the number of sub-events of a particular duration that may be derived from an event. Let $d_c \in \mathbb{N}$ be the duration of the sub-events for which this constraint applies, and let \underline{B}_c and \overline{B}_c be the minimum and maximum number of sub-events of duration d_c that may be derived from a given event, respectively.

$$\underline{B}_c - \sum_{\substack{se \in \mathcal{SE}_e \\ D_{se}=d_c}} u_{se} \leq s_{c,e}^4 \quad \forall c \in \mathcal{C}^4, \forall e \in \mathcal{E}_c \quad (20)$$

$$\sum_{\substack{se \in \mathcal{SE}_e \\ D_{se}=d_c}} u_{se} - \overline{B}_c \leq s_{c,e}^4 \quad \forall c \in \mathcal{C}^4, \forall e \in \mathcal{E}_c \quad (21)$$

\mathcal{C}^5 – **Prefer Resources** : This constraint defines that an event resource has a preference for certain resources. The assignment of all non-preferred resources is taken and the duration of the sub-events in which these non-preferred resources are assigned is summed to calculate the deviation.

Let $r \in \mathcal{R}_c$ denote a preferred resource:

$$\sum_{se \in \mathcal{SE}_e} \sum_{r \in \mathcal{R} \setminus \{r_D\} : r \notin \mathcal{R}_c} D_{se} \times w_{se,er,r} = s_{c,er}^5 \quad \begin{array}{l} \forall c \in \mathcal{C}^5 \\ \forall e \in \mathcal{E}_c \\ \forall er \in \mathcal{ER}_e : PA_{er}=0 \wedge role_{er}=role_c \end{array} \quad (22)$$

\mathcal{C}^6 – **Prefer Times** : Like the prefer resources constraint, events might also have preferences for certain times. The deviation is calculated for each event by summing the duration of all sub-events which are assigned a time that is not in the preferred times list. The constraint has an optional duration-property, denoted $D_c \in \mathbb{N}_0$. If this property is given, only sub-events of duration D_c are considered. Let $t \in \mathcal{T}_c$ denote a preferred time:

$$\sum_{se \in \mathcal{SE}_e} \sum_{t \in \mathcal{T} \setminus \{t_D\} : t \notin \mathcal{T}_c \wedge D_c = D_{se}} D_{se} \times y_{se,t} = s_{c,er}^6 \quad \begin{array}{l} \forall c \in \mathcal{C}^6 \\ \forall e \in \mathcal{E}_c \end{array} \quad (23)$$

\mathcal{C}^7 – **Avoid Split Assignments** : When an event is split into sub-events, each of its event resources is also split for each sub-event. A different resource may be assigned to each of these generated event resources. This constraint penalizes the assignment of different resources to event resources within the same event group. The constraint examines the demand of all event resources derived from the events in the event group and it calculates the number of distinct resources assigned to them, ignoring unassigned event resources. The deviation $s_{c,eg}^7$ is the number of resources that exceeds 1.

$$\sum_{\substack{er \in \mathcal{ER}_e, PA_{er}=0 \\ role_c = role_{er}}} w_{se,er,r} \leq k_{eg,r} \quad \begin{array}{l} \forall c \in \mathcal{C}^7 \\ \forall r \in \mathcal{R} \\ \forall eg \in \mathcal{EG}_c \\ \forall e \in \mathcal{E}_{eg} \\ \forall se \in \mathcal{SE}_e \end{array} \quad (24)$$

$$\sum_{r \in \mathcal{R}} k_{eg,r} - 1 \leq s_{c,eg}^7 \quad \begin{array}{l} \forall c \in \mathcal{C}^7 \\ \forall eg \in \mathcal{EG}_c \end{array} \quad (25)$$

\mathcal{C}^8 – **Spread Events** : The spread events constraint has a deviation for each time group $tg \in \mathcal{TG}_c \in \mathcal{C}^8$. Let $\underline{B}_{c,tg}$ and $\overline{B}_{c,tg}$ be, respectively, the minimum and maximum number of sub-events of a given event that can be placed in time group tg of constraint c . The deviation $s_{c,eg,tg}^8$

for each time group is given by the number of sub-events for the given event that is below $\underline{B}_{c,tg} \in \mathbb{N}$ or above $\overline{B}_{c,tg} \in \mathbb{N}$.

$$\underline{B}_{c,tg} - \sum_{se \in \mathcal{SE}_e} \sum_{e \in \mathcal{EG}_c} \sum_{t \in \mathcal{T}_{tg}} y_{se,t} \leq s_{c,eg,tg}^8 \quad \begin{array}{l} \forall c \in \mathcal{C}^8 \\ \forall eg \in \mathcal{EG}_c \\ \forall tg \in \mathcal{TG}_c \end{array} \quad (26)$$

$$\sum_{se \in \mathcal{SE}_e} \sum_{e \in \mathcal{EG}_c} \sum_{t \in \mathcal{T}_{tg}} y_{se,t} - \overline{B}_{c,tg} \leq s_{c,eg,tg}^8 \quad \begin{array}{l} \forall c \in \mathcal{C}^8 \\ \forall eg \in \mathcal{EG}_c \\ \forall tg \in \mathcal{TG}_c \end{array} \quad (27)$$

\mathcal{C}^9 – **Link Events** : A link event constraint specifies that a set of events within an event group should be assigned to the same starting time. The deviation of this constraint is set as the number of times in which at least one event in the event group does not occur simultaneously with the others. Constraints (28), (29) and (30) ensure that variables $o_{e,t}$ and $l_{eg,t}$ assume correct values. The slack variable of Link Events constraint, $s_{c,eg,t}^9$, is defined in constraint (31).

$$\sum_{t' \in \mathcal{T}_{se,t}^{start}} y_{se,t'} \leq o_{e,t} \quad \begin{array}{l} \forall e \in \mathcal{E} \\ \forall se \in \mathcal{SE}_e \\ \forall t \in \mathcal{T} \setminus \{t_D\} \end{array} \quad (28)$$

$$\sum_{se \in \mathcal{SE}_e} \sum_{t' \in \mathcal{T}_{se,t}^{start}} y_{se,t'} \geq o_{e,t} \quad \begin{array}{l} \forall e \in \mathcal{E} \\ \forall se \in \mathcal{SE}_e \\ \forall t \in \mathcal{T} \setminus \{t_D\} \end{array} \quad (29)$$

$$l_{eg,t} \geq o_{e,t} \quad \begin{array}{l} \forall eg \in \mathcal{EG} \\ \forall e \in \mathcal{EG}_c \\ \forall t \in \mathcal{T} \setminus \{t_D\} \end{array} \quad (30)$$

$$l_{eg,t} - o_{e,t} \leq s_{c,eg,t}^9 \quad \begin{array}{l} \forall c \in \mathcal{C}^9 \\ \forall eg \in \mathcal{EG}_c \\ \forall t \in \mathcal{T} \setminus \{t_D\} \end{array} \quad (31)$$

\mathcal{C}^{10} – **Order Events** : An order events constraint specifies that the times assigned to two events should be in order, in such a way that the first event ends before the second event starts. Let parameter $\underline{B}_c \in \mathbb{N}$ and $\overline{B}_c \in \mathbb{N}$ be, respectively, the minimum and maximum number of times that may separate two events. Let (e, \hat{e}) denote an event pair such that this constraint applies to. The deviation, $s_{c,e,\hat{e}}^{10}$, is then given by the amount by which the difference between these h_e^{last} and $h_{\hat{e}}^{first}$ exceeds

\overline{B}_c or falls below \underline{B}_c .

$$p_t \times y_{se,t} + D_{se} \leq h_e^{last} \quad \begin{array}{l} \forall c \in \mathcal{C}^{10} \\ \forall e \in \mathcal{E}_c \\ \forall se \in \mathcal{SE}_e \\ \forall t \in \mathcal{T} \setminus \{t_D\} \end{array} \quad (32)$$

$$|\mathcal{T}| - (|\mathcal{T}| - p_t) \times y_{se,t} \geq h_e^{first} \quad \begin{array}{l} \forall c \in \mathcal{C}^{10} \\ \forall e \in \mathcal{E}_c \\ \forall se \in \mathcal{SE}_e \\ \forall t \in \mathcal{T} \setminus \{t_D\} \end{array} \quad (33)$$

$$\underline{B}_c - (h_{\hat{e}}^{first} - h_e^{last}) \leq s_{c,e,\hat{e}}^{10} \quad \begin{array}{l} \forall c \in \mathcal{C}^{10} \\ \forall (e,\hat{e}) \in (E,E)_c \end{array} \quad (34)$$

$$(h_{\hat{e}}^{first} - h_e^{last}) - \overline{B}_c \leq s_{c,e,\hat{e}}^{10} \quad \begin{array}{l} \forall c \in \mathcal{C}^{10} \\ \forall (e,\hat{e}) \in (E,E)_c \end{array} \quad (35)$$

\mathcal{C}^{11} – **Avoid Clashes** : These constraints specify that certain resources should not have clashes in their timetables. It means they should not be assigned to two or more events simultaneously. This constraint produces a set of deviations for each resource. For each time, the number of occurrences of given resource minus one is calculated to estimate the deviation of that resource for that time.

$$v_{t,r} - 1 \leq s_{c,r,t}^{11} \quad \begin{array}{l} \forall c \in \mathcal{C}^{11} \\ \forall r \in \mathcal{R}_c \\ \forall t \in \mathcal{T} \setminus \{t_D\} \end{array} \quad (36)$$

\mathcal{C}^{12} – **Avoid Unavailable Times** : An avoid unavailable times constraint specifies that certain resources are unavailable for any event at certain times. The deviation, $s_{c,r}^{12}$ is the number of times that are being attended by an unavailable resource. Let $t \in \mathcal{T}_c$ denote that t is an unavailable time for constraint $c \in \mathcal{C}^{12}$:

$$\sum_{t \in \mathcal{T}_c} q_{r,t} = s_{c,r}^{12} \quad \begin{array}{l} \forall c \in \mathcal{C}^{12} \\ \forall r \in \mathcal{R}_c \end{array} \quad (37)$$

\mathcal{C}^{13} – **Limit Idle Times** : A resource is idle at some time $t \in \mathcal{T}_{tg}$ if it is not attending any sub-event at t , but it is attending events before and after t in the same time group tg . Limit idle times constraint limits the number of idle times a resource may have within a time group.

$$q_{r,\hat{t}} - q_{r,t} + q_{r,\hat{t}} - 1 \leq h_{r,tg,t} \quad \begin{array}{l} \forall r \in \mathcal{R} \\ \forall tg \in \mathcal{TG} \\ \forall t, \hat{t} \in \mathcal{T}_{tg} : p_{\hat{t}} < p_t < p_{\hat{t}} \end{array} \quad (38)$$

For each resource of the constraint, the deviation is calculated as fol-

lows: sum the number of cases in which the amount of idle times is under $\underline{B}_c \in \mathbb{N}$ or above $\overline{B}_c \in \mathbb{N}$. Slack variable $s_{c,r}^{13}$ computes the sum of such cases.

$$\underline{B}_{c,r} - \sum_{tg \in \mathcal{TG}_c} h_{r,tg} \leq s_{c,r}^{13} \quad \forall c \in \mathcal{C}^{13} \quad \forall r \in \mathcal{R}_c \quad (39)$$

$$\sum_{tg \in \mathcal{TG}_c} h_{r,tg} - \overline{B}_{c,r} \leq s_{c,r}^{13} \quad \forall c \in \mathcal{C}^{13} \quad \forall r \in \mathcal{R}_c \quad (40)$$

\mathcal{C}^{14} – **Cluster Busy Times** : A cluster busy times constraint limits the number of time groups in which a resource may be busy. The deviation is given by number of cases in which the resource is busy for less than $\underline{B}_c \in \mathbb{N}$ time groups, or for more than $\overline{B}_c \in \mathbb{N}$ time groups. Let $tg \in \mathcal{TG}_c$ denote a time group in which such a constraint applies:

$$\underline{B}_c - \sum_{tg \in \mathcal{TG}_c} p_{r,tg} \leq s_{c,r}^{14} \quad \forall c \in \mathcal{C}^{14} \quad \forall r \in \mathcal{R}_c \quad (41)$$

$$\sum_{tg \in \mathcal{TG}_c} p_{r,tg} - \underline{B}_c \leq s_{c,r}^{14} \quad \forall c \in \mathcal{C}^{14} \quad \forall r \in \mathcal{R}_c \quad (42)$$

\mathcal{C}^{15} – **Limit Busy Times** : Limit busy times constraint places limits on the number of times a resource may be busy within some time groups. These constraints produce a deviation for each time group. The deviations are given by the number of cases in which the resource is busy for less than $\underline{B}_c \in \mathbb{N}$ time groups, or for more than $\overline{B}_c \in \mathbb{N}$ time groups.

$$\underline{B}_c - |\mathcal{T}_{tg}| \times (1 - p_{r,tg}) - \sum_{t \in \mathcal{T}_{tg}} q_{r,t} \leq s_{c,r,tg}^{15} \quad \forall c \in \mathcal{C}^{15} \quad \forall r \in \mathcal{R}_c \quad \forall tg \in \mathcal{TG}_c \quad (43)$$

$$\sum_{t \in \mathcal{T}_{tg}} q_{r,t} - \overline{B}_c - |\mathcal{T}_{tg}| \times (1 - p_{r,tg}) \leq s_{c,r,tg}^{15} \quad \forall c \in \mathcal{C}^{15} \quad \forall r \in \mathcal{R}_c \quad \forall tg \in \mathcal{TG}_c \quad (44)$$

\mathcal{C}^{16} – **Limit Workload** : The workload of a resource is given by $W_{e,se,er} = \frac{D_{se} \times L_{er}}{D_e}$, in which $L_{er} \in \mathbb{N}$ is the workload of the event resource er . These values are given as inputs within the information related to events. A limit workload constraint places limits on the total workload that is assigned to resources. The deviation of this constraint, $s_{c,r}^{16}$, is the amount of cases in which the resource workload falls short

$\underline{B}_c \in \mathbb{N}$, or exceeds $\underline{B}_c \in \mathbb{N}$, rounded up to the nearest integer.

$$\underline{B}_c - \sum_{e \in \mathcal{E}_c} \sum_{t \in \mathcal{T} \setminus \{t_D\}} \sum_{se \in \mathcal{SE}_e} \sum_{er \in \mathcal{ER}_e} W_{e,se,er} \times x_{se,t,er,r} \leq s_{c,r}^{16} \quad \forall c \in \mathcal{C}^{16} \quad \forall r \in \mathcal{R}_c \quad (45)$$

$$\sum_{e \in \mathcal{E}_c} \sum_{t \in \mathcal{T} \setminus \{t_D\}} \sum_{se \in \mathcal{SE}_e} \sum_{er \in \mathcal{ER}_e} W_{e,se,er} \times x_{se,t,er,r} - \overline{B}_c \leq s_{c,r}^{16} \quad \forall c \in \mathcal{C}^{16} \quad \forall r \in \mathcal{R}_c \quad (46)$$

2.3. Objective Function

Given the slack variables for each XHSTT constraint, the cost of a point-of-application of a constraint $c \in \mathcal{C}$ is defined based on three properties: *type_c* (it can be either *hard* or *soft*), weight ($w_c \in \mathbb{N}$), and the *CostFunction* (CF) to use.

The cost of a constraint $c \in \mathcal{C}$, which contains slack variable $s_{c,p,d}$, is denoted by $f(s_{c,p,d})$, and it is calculated as shown in (47).

$$f(s_{c,p,d}) = w_c \times \text{CostFunction}(s_{c,p,d}). \quad (47)$$

Three different cost function types are allowed: linear, quadratic and step. These functions are evaluated in terms of slack variables $s_{c,p,d}$, such as follows:

Linear : Sum of deviations.

$$CF^{Linear} = \sum_{p \in \mathcal{P}_c} \sum_{d \in \mathcal{D}_p} s_{c,p,d}. \quad (48)$$

Quadratic : Sum of deviation squares.

A variable $i \in \{0, 1\}$ is introduced to handle with this non-linear cost function. It assumes value 1 if the deviation $d \in \mathcal{D}_p$ of the point of application $p \in \mathcal{P}_c$ of constraint c has the value $i \in \mathcal{I}$, or 0 otherwise. Let $\mathcal{I} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$:

$$CF^{Quadratic} = \sum_{p \in \mathcal{P}_c} \sum_{d \in \mathcal{D}_p} \sum_{i \in \mathcal{I}} i^2 \times s_{c,d,p,i} \quad (49)$$

The link from $s_{c,p,d}$ to $s_{c,p,d,i}$ is given by:

$$\sum_{p \in \mathcal{P}_c} \sum_{d \in \mathcal{D}_p} \sum_{i \in \mathcal{I}} i \times s_{c,d,p,i} = s_{c,p,d} \quad \begin{matrix} \forall p \in \mathcal{P}_c \\ \forall d \in \mathcal{D}_p \end{matrix} \quad (50)$$

It is also necessary to ensure that only a single integer value is selected by the binary indicator. Therefore the following constraint set must be considered:

$$\sum_{i \in \mathcal{I}} i \times s_{c,d,p,i} = 1 \quad \begin{matrix} \forall p \in \mathcal{P}_c \\ \forall d \in \mathcal{D}_p \end{matrix} \quad (51)$$

Step : Penalizes the number of deviations, regardless their magnitudes.

The binary variable $u_{c,p,d}^{Step}$ is introduced. It assumes value 1 iff $s_{c,p,d} > 0$ for constraint c , point-of-application $p \in \mathcal{P}_c$ and deviation $d \in \mathcal{D}_p$, or 0 otherwise. The cost function and constraint set are formulated as shown in (52) and (53), respectively.

$$CF^{Step} = \sum_{p \in \mathcal{P}_c} \sum_{d \in \mathcal{D}_p} u_{c,p,d}^{Step} \quad (52)$$

$$M \times u_{c,p,d}^{Step} \geq s_{c,p,d} \quad \begin{matrix} \forall p \in \mathcal{P}_c \\ \forall d \in \mathcal{D}_p \end{matrix} \quad (53)$$

in which $M \in \mathbb{N}$ is a sufficiently large number.

Let Ψ_c be the set of tuples that a constraint $c \in C$ applies to. For example, suppose $c \in C^4$, then for a $\tau \in \Psi_c$, $\tau \equiv (e, er)$ and $s_{c,\tau}^4 \equiv s_{c,e,er}^4$. Thus, the objective function can be stated as shown in equation (54).

$$z = \sum_{j=1}^{16} \sum_{c \in C^j} \sum_{\tau \in \Psi_c} f(s_{c,\tau}^j) \quad (54)$$

The full IP model consists on minimizing z , subject to constraints (2) to (54). This model is solved in two steps. In Step 1, an IP model is built containing only the hard constraints. When the IP solver finds an optimal solution regarding this first model (hopefully a zero-cost solution), all soft constraints are included to the model and the solution process is warm-started from its previous state (Step 2). Furthermore, a constraint is added to ensure that the optimal value of hard cost is kept unchanged:

$$z^{hard} = \text{hard cost} \quad (55)$$

in which z^{hard} denotes the sum of the cost of all slack variables associated to hard constraints.

Once such an IP model is solved, the cost of the obtained solution, minus the hard cost found in Step 1, is the penalty due to soft constraint violations.

3. Alternative Formulation

In this section, starting from the original formulation \mathcal{F}_1 presented in the previous section, we propose some valid inequalities and an extended flow based formulation for XHSTT, yielding an improved formulation denoted throughout the paper as \mathcal{F}_2 . As it can be seen in Section 4, \mathcal{F}_2 is stronger than \mathcal{F}_1 , since it is strictly contained in \mathcal{F}_1 , i.e. valid fractional points for \mathcal{F}_1 are not valid for \mathcal{F}_2 . These cuts were found either by analysing fractional solutions of \mathcal{F}_1 (3.5 and 3.7) or with the aid of an automatic integer programming reformulation tool [24] (3.4 and 3.6), which searches for fractional multipliers to generate Chvátal-Gomory cuts [25]. This is a computationally expensive, offline tool. Once these multipliers were discovered, it was relatively easy to generalize these cuts to the timetabling context, such that now their separation is trivial. Pre-processing routines are also proposed to remove unnecessary constraints and variables. All those changes are described in the following subsections.

3.1. Generation of Sub-events

In \mathcal{F}_2 , only feasible sub-events are generated. A sub-event is called feasible when it does not violate any hard Split Events constraint. This reduction aims to make the resulting IP model smaller, faster to build, and less dependant on the duration of the events. For example, suppose an event e of duration $D_e = 4$ and a hard Split Events constraints c stating that e should be split into sub-events of duration 2 ($\overline{B}_c^{dur} = \underline{B}_c^{dur} = 2$). In the complete formulation, the following set of sub-events having durations $\mathcal{SE}_e = \{1, 1, 1, 1, 2, 2, 3, 4\}$ would be generated. In this reduced reformulation, the resulting set of sub-events would be considerably smaller: $\mathcal{SE}_e = \{2, 2\}$. However, this benefit is achieved at expense of not ensuring optimality in a special case in which the optimal solution has hard cost different from zero. In practice, this is not a big problem for two reasons: (i) most of XHSTT problems are feasible (i.e. they have at least one possible solution with hard cost equals to zero), and; (ii) in real world problems, it is expected to exist a solution having zero cost of hard constraints.

Algorithm 1: Alternative procedure to generate sub-events

Input: A set of events \mathcal{E} , a set of split events constraints \mathcal{C}^3 .

Output: A set of sub-events \mathcal{SE} per event $e \in \mathcal{E}$.

```
1  $\mathcal{SE} = \emptyset$ ;  
2 foreach  $e \in \mathcal{E}$  do  
3    $\mathcal{SE}_e = \emptyset$ ;  
4   foreach  $k = \{1, \dots, D_e\}$  do  
5     if  $\nexists c \in \mathcal{C}^3 \mid e \in \mathcal{E}_c, \text{ type}_c = \text{hard}$  then  
6       foreach  $j = \{k, \dots, D_e\}$  do  
7          $D_{se} = k$ ;  
8          $\mathcal{SE}_e = \mathcal{SE}_e \cup \{D_{se}\}$ ;  
9     else  
10      if  $k \geq \overline{B}_c^{dur}$  and  $k \leq \underline{B}_c^{dur}$  then  
11         $subeventsAmount = 0$ ;  
12        foreach  $j = \{k, \dots, D_e\}$  do  
13          if  $subeventsAmount \leq \overline{B}_c^{amt}$  then  
14             $D_{se} = k$ ;  
15             $\mathcal{SE}_e = \mathcal{SE}_e \cup \{D_{se}\}$ ;  
16    $\mathcal{SE} = \mathcal{SE} \cup \mathcal{SE}_e$ ;  
17 return  $\mathcal{SE}$ ;
```

3.2. Alternative Formulation for Link Events

One of the constraints that makes this problem difficult is the requirement of linked events. In this alternative formulation, the structure of Link Events constraints is explored. When events are connected by hard Link Events constraints, they must occur at the same time and they must be split in the same way. Considering this fact, constraints (28), (29), (30), and (31) can be replaced by:

$$y_{se,t} = y_{\hat{se},t} \quad \begin{array}{l} \forall c \in \mathcal{C}^{10}, eg \in \mathcal{EG}_c, \\ e \in eg \mid first(eg) = e, \\ \hat{e} \in eg \mid first(eg) \neq \hat{e}, \\ (se, \hat{se}) \in (\mathcal{SE}_e, \mathcal{SE}_{\hat{e}}), t \in T \setminus t_D \end{array} \quad (56)$$

in which $first(\cdot)$ returns the first element of the set of events in event group eg . When the events that belong to a same event group constrained by Link Events have different durations or are constrained by different split events constraints this reformulation is not applied. Once again, such a reformulation does not ensure optimality if the the optimal solution has hard cost different from zero.

3.3. Alternative Formulation for Avoid Clashes

In real world problems, as well as for all the existing XHSTT instances, the Avoid Clashes constraints are always hard ones. Therefore, constraints (4), (11), (12), and (36) can be replaced by:

$$\sum_{se \in \mathcal{SE}} \sum_{er \in \mathcal{ER}_{se}} \sum_{\hat{t} \in \mathcal{T}_{se,t}^{start}} x_{se,\hat{t},er,r} = q_{r,t} \quad \begin{array}{l} \forall t \in \mathcal{T} \setminus \{t_D\} \\ \forall r \in \mathcal{R} \end{array} \quad (57)$$

Note that the auxiliary variable type $v_{t,r} \in \mathbb{N}$ is no longer used in \mathcal{F}_2 . Consequently, it was removed from the model in the alternative formulation.

3.4. Alternative Formulation to Link X and Y (LXY)

The link between variables $x_{se,t,er,r}$ and $y_{se,t}$ in \mathcal{F}_1 might be strengthened if one considers the link of each single variable $x_{se,t,er,r}$ to each single variable $y_{se,t}$, instead of linking a set of variables $x_{se,t,er,r}$ to $|er_{se}| \times y_{se,t}$. Furthermore, when a hard Prefer Resources constraint applies to event e that generates sub-event se , only the preferred resources should be eligible for the assignment (assuming the existence of a feasible solution for the problem). Taking into account such points, constraint (3) was replaced by:

$$x_{se,t,er,r} = y_{se,t} \quad \begin{array}{l} \forall se \in \mathcal{SE} \\ \forall t \in \mathcal{T}, \\ \forall er \in \mathcal{ER}_{se}, \\ \forall r \in \mathcal{R}_{er} : r \in \mathcal{C}^5 \vee \mathcal{C}^5 = \emptyset \vee type_c \neq hard \end{array} \quad (58)$$

3.5. Cluster Busy Times Cut (CBT)

The number of days a resource may be busy is at least the sum of the durations of the events that it is preassigned to divided by the number of times per day *timesDay*. This number is rounded up to the nearest larger integer. This cut was first proposed by Santos *et al.* [18] and adapted to XHSTT in this work:

$$\sum_{tg \in \mathcal{TG}_c} p_{r,tg} \geq \left\lceil \frac{\sum_{\substack{e \in \mathcal{E}: \\ er \in \mathcal{ER}_e \\ PR_{er,r}=1}} D_e}{timesDay} \right\rceil \quad \begin{array}{l} \forall c \in \mathcal{C}^{14} \\ \forall r \in \mathcal{R}_c \end{array} \quad (59)$$

3.6. Link Y and Q Cut (LYQ)

For any resource r and time t , if r is busy at t , at least one of the sub-events that have r as a preassigned resource will be occurring at t . If a hard Prefer Resources constraint applies to the sub-event, the sub-event will be considered only when the resource is a preferred resource.

$$q_{r,t} \leq \sum_{\substack{se \in \mathcal{SE} \\ r \in er \in \mathcal{ER}_{se} \wedge \\ r \in \mathcal{C}^5 \vee \mathcal{C}^5 = \emptyset \vee type_c \neq hard}} y_{se,t} \quad \begin{array}{l} \forall r \in \mathcal{R} \\ \forall t \in \mathcal{T} \setminus \{t_D\} \end{array} \quad (60)$$

3.7. Number of Busy Times Cut (NBT)

The number of busy times of a resource r is computed through variables $q_{r,t}$. However, in order to strengthen the formulation, this number could be explicitly given to the IP model when no resource assignment is required:

$$\sum_{t \in \mathcal{T}} q_{r,t} = \sum_{\substack{e \in \mathcal{E}: \\ er \in \mathcal{ER}_e \wedge PR_{er,r}=1}} D_e \quad \forall r \in \mathcal{R} \quad (61)$$

If an Assign Resource constraint is considered, the inequality, for any resource eligible to more assignments than the preassigned ones is given by:

$$\sum_{t \in \mathcal{T}} q_{r,t} \geq \sum_{\substack{e \in \mathcal{E}: \\ er \in \mathcal{ER}_e \wedge PR_{er,r}=1}} D_e \quad \begin{array}{l} \forall r \in \mathcal{R}: \\ \exists c \in \mathcal{C}^1 \mid type_r = roleType(role_c) \end{array} \quad (62)$$

3.8. Multicommodity Flow Reformulation (MCF)

Dorneles *et al.* [26] modelled the High School Timetabling problem as a Multicommodity Flow problem. Following, an adaptation of Dorneles' model to formulate some constraints of XHSTT is presented.

Each resource is represented by a commodity. For each resource $r \in \mathcal{R}$ a graph whose flow represents the resource schedules is created. Let $\mathcal{G}_r = (\mathcal{V}_r, \mathcal{A}_r)$ denote such a graph, in which \mathcal{V} is the set of nodes and \mathcal{A} is the set of arcs. Each node has a set of pull in arcs $\mathcal{A}_{r,v}^+ \subseteq \mathcal{A}$ and a set of pull out arcs $\mathcal{A}_{r,v}^- \subseteq \mathcal{A}$. Variables

$$\varphi_a = \begin{cases} 1 & \text{if flow goes through arc } a \\ 0 & \text{otherwise} \end{cases}$$

are added to the new formulation \mathcal{F}_2 and parameter b_v has value 1 when node v is the source, -1 when it is the sink, or 0 otherwise.

Figure 1 presents an example of this graph, in which all types of arcs are shown for a given resource r . Each arc has a specific meaning. Whenever the arc has the same meaning of a variable in \mathcal{F}_1 , new variable φ_a is not created and the respective variable in \mathcal{F}_1 is used instead. The meaning of each type of arc $a \in \mathcal{A}$ is given below:

- Assignment arcs are used to denote that a given resource r is attending one event at time t . These arcs are denoted by binary variables $q_{r,t}$, whose meaning is the same described before in the complete formulation.
- Idle time arcs represent that a resource r has an idle time between busy times at time t in time group tg . These arcs correspond to binary variable $h_{r,tg,t}$, which also came from the original formulation.
- Cluster busy time arcs denote that a given resource r is busy for at least one time in time group tg . These arcs are denoted by binary variables $p_{r,tg}$, which are also the same from the complete formulation.
- Arcs $a_{r,tg,t}^{IN}$ and $a_{r,tg,t}^{OUT}$ are given for each resource r , time group tg and time t . They denote, respectively, that the first assignment in time group tg for resource r is at time t and that the last assignment is at time t .

- Day-off arcs $a_{r,tg}^{OFF}$ represent that a resource r is not busy at any of the times in time group tg . These arcs lead to a node that represents the next time group or to the sink node.

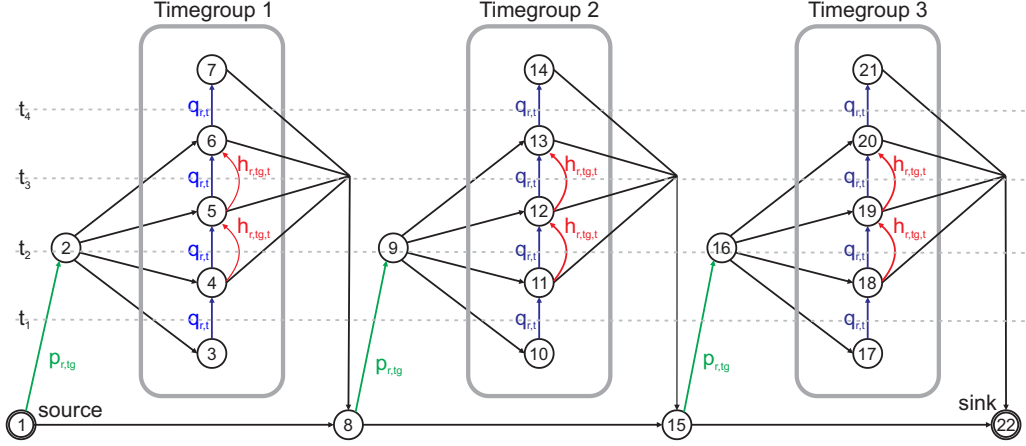


Figure 1: Example of network for a resource in a toy instance consisting of three days, having four times each (adapted from [26]).

Every path in such a graph starts from the source node, alternates the arcs providing information about the time assignments, idle times, and busy time groups (usually days) for the resource and ends at the sink node. Figure 2 presents an example of a feasible flow for a given resource. In this example, the resource is busy at the first time (t_1) of Timegroup 1, has a idle time in t_2 and is busy again in times t_3 and t_4 ; then the resource has a day off in Timegroup 2; and the resource is busy again at times t_2 and t_3 in Timegroup 3.

The following set of constraints is added to the formulation to ensure the flow conservation in the nodes:

$$\sum_{a \in \mathcal{A}_{r,v}^+} \varphi_a + \sum_{a \in \mathcal{A}_{r,v}^-} \varphi_a = b_v \quad \forall r \in \mathcal{R} \quad \forall v \in V_r \quad (63)$$

Some paths should be explicitly forbidden on the network flow formulation since they lead to miss calculation of penalties. Figure 3 illustrates the two cases that shall be avoided. In the first case, an arc goes to node 5 and another leaves right from node 5. This case leads to miss calculation of

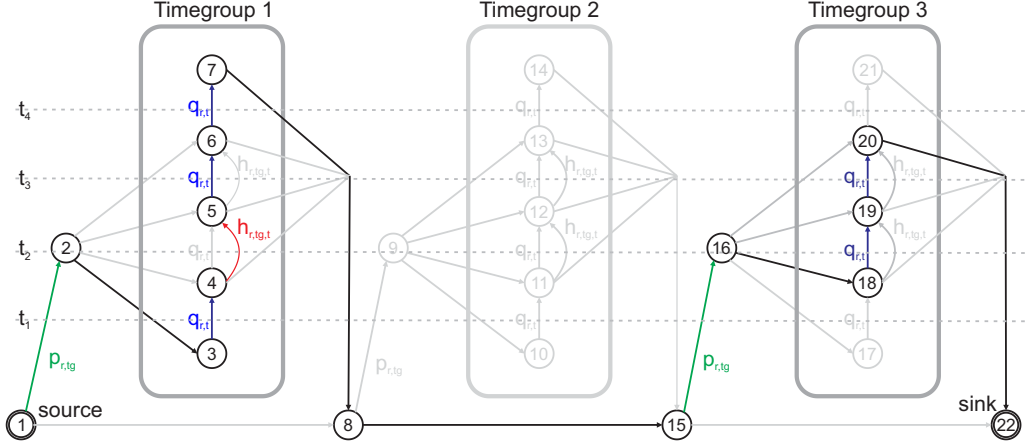


Figure 2: Example of schedule for one resource r (adapted from [26]).

penalties for Cluster Busy Times constraints because the flow goes into a time group even if it does not have any assignment in any time of that day. Therefore a time group would be counted as busy when it is in fact not busy. Constraint (64) is added to forbid such paths:

$$a_{r,tg,t}^{IN} + a_{r,tg,t}^{OUT} \leq 1 \quad \begin{matrix} \forall r \in \mathcal{R} \\ \forall tg \in \mathcal{TG} \\ \forall t \in \mathcal{T}_{tg} \end{matrix} \quad (64)$$

In the second case, constraint Limit Idle Times is miss calculated because a time should not be counted as idle when there is no busy time before it in a given time group. The time also should not be counted when there is no busy time after itself within the same time group. Constraints (65) and (66) disable such paths:

$$a_{r,tg,t}^{IN} + h_{r,tg,t} \leq 1 \quad \begin{matrix} \forall r \in \mathcal{R} \\ \forall tg \in \mathcal{TG} \\ \forall t \in \mathcal{T}_{tg} \end{matrix} \quad (65)$$

$$a_{r,tg,t}^{OUT} + h_{r,tg,t} \leq 1 \quad \begin{matrix} \forall r \in \mathcal{R} \\ \forall tg \in \mathcal{TG} \\ \forall t \in \mathcal{T}_{tg} \end{matrix} \quad (66)$$

Constraints Cluster Busy Times and Limit Idle Times are handled by this network flow model. Therefore constraints (13), (14) and (38) can be removed from \mathcal{F}_2 .

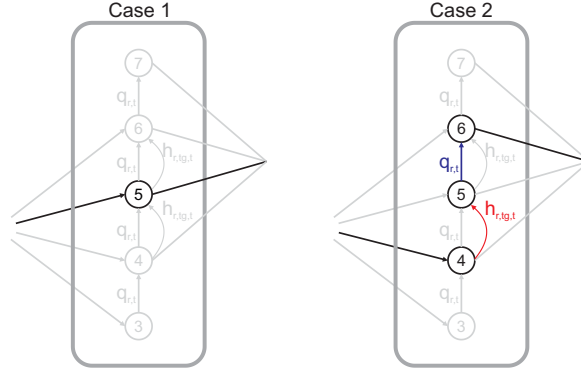


Figure 3: Forbidden paths in the network (adapted from [26]).

4. Computational Experiments

Experiments were performed on an Intel[®] i7 4510-U 2.6 Ghz PC with 8GB of RAM under Ubuntu 14.04 operating system. The software was coded in C++ and compiled with GCC 4.6.1. The obtained results were validated with the HSEval validator¹. The state-of-the-art solver Gurobi 6.5.1, with default parameter settings, was used to solve the mathematical programming models.

The results are represented by the pair (H, S) , in which H and S denote the cost of hard and soft constraint violations, respectively. When hard-constraints are not violated, only the cost of soft-constraint violations is reported. Our solver, along with our solutions and reports, can be found at the GOAL-UFOP website². We invite the interested reader to validate our results.

4.1. Characterization of Instances

The ITC2011 Hidden³ (XHSTT-ITC2011-hidden) instance set was considered to compare the formulations. This set covers features of timetabling from several countries and has variate problem sizes, ranging from instances demanding the assignment of 75 events to instances that demand the assignment of thousands of events and resources. Moreover, this instance set is

¹<http://sydney.edu.au/engineering/it/ and ~jeff/hseval.cgi>

²<http://www.goal.ufop.br/software/hstt>

³<https://www.utwente.nl/ctit/hstt/archives/XHSTT-ITC2011-hidden/>

currently the most used to evaluate algorithms for XHSTT in the literature. Table 1 presents the occurrence of constraints and whether they are hard or soft in each instance. A hard constraint is denoted as H and a soft constraint is identified as S. An empty cell means that the constraint is not considered on that instance.

Table 1: Constraints in XHSTT-ITC2011-hidden instances.

Instance	Assign Times	Assign Resources	Prefer Times	Prefer Resources	Link Events	Spread Events	Avoid Split Assignments	Distribute Split Events	Split Events	Avoid Clashes	Avoid Unavailable Times	Limit Workload	Limit Idle Times	Limit Busy Times	Cluster Busy Times
BrazilInstance2	H		S			H		S	H	H	H		S		S
BrazilInstance3	H		S			H		S	H	H	H		S		S
BrazilInstance4	H		S			H		S	H	H	H		S		S
BrazilInstance6	H		S			H		S	H	H	H		S		S
FinlandElementarySchool	H		H			S			H	H	H			S	
FinlandSecondarySchool2	H					H			H	H			S	S	
Aigio1stHighSchool10-11	H		S		H	H			H	H	H		S	S	
ItalyInstance4	H		H			H			H	H	H		S	S	
KosovaInstance1	H		H			H		S	H	H	H		H	H	
Kottenpark2003	H	H	H	H	H	H			H	H	H		S	S	S
Kottenpark2005A	H	H	H	H	H	H			H	H	H		S	S	S
Kottenpark2008	H	H	H	H	H	H			H	H	H		H		S
Kottenpark2009	H	H	H	H	H	H			H	H	H		S		S
Woodlands2009	H		S		S					H	S				
Spanish school	H	H		H		H			H	H	H			S	S
WesternGreeceUniversity3	H					S				H	H		H	H	
WesternGreeceUniversity4	H				H	S				H	H		H	H	
WesternGreeceUniversity5	H					S				H	H		H	H	

4.2. Cut Effectiveness

Table 2 presents, for each cut, whether its addition to \mathcal{F}_1 changes the optimal solution of the linear relaxation of the model or not. Each entry in

the table represents the execution of the linear relaxation of \mathcal{F}_1 applying the refereed cut on it. The cuts were ordered from the one that was effective to most instances to the one that was effective to least instances. The following abbreviations were used: Number of Busy Times Cut (NBT); Link X and Y Cut (LXY); Multicommodity Flow (MCF); Link Y and Q Cut (LYQ); and Cluster Busy Times Cut (CBT).

Table 2: Effectiveness of each cut over XHSTT-ITC2011-hidden archive					
Instance	NBT	LXY	MCF	LYQ	CBT
BrazilInstance2	✓	✓	✓		✓
BrazilInstance3	✓	✓	✓		✓
BrazilInstance4	✓	✓	✓		✓
BrazilInstance6	✓	✓	✓		✓
FinlandElementarySchool	✓	✓			
FinlandSecondarySchool2	✓	✓	✓		
Aigio1stHighSchool10-11	✓	✓	✓	✓	
ItalyInstance4	✓	✓	✓	✓	
KosovaInstance1	✓	✓	✓		
Kottenpark2003	✓	✓	✓	✓	✓
Kottenpark2005A	✓	✓	✓	✓	✓
Kottenpark2008	✓	✓	✓	✓	✓
Kottenpark2009	✓	✓	✓	✓	✓
Woodlands2009	✓	✓		✓	
Spanish school	✓	✓		✓	
WesternGreeceUniversity3	✓	✓	✓		
WesternGreeceUniversity4	✓	✓	✓	✓	
WesternGreeceUniversity5	✓	✓	✓		

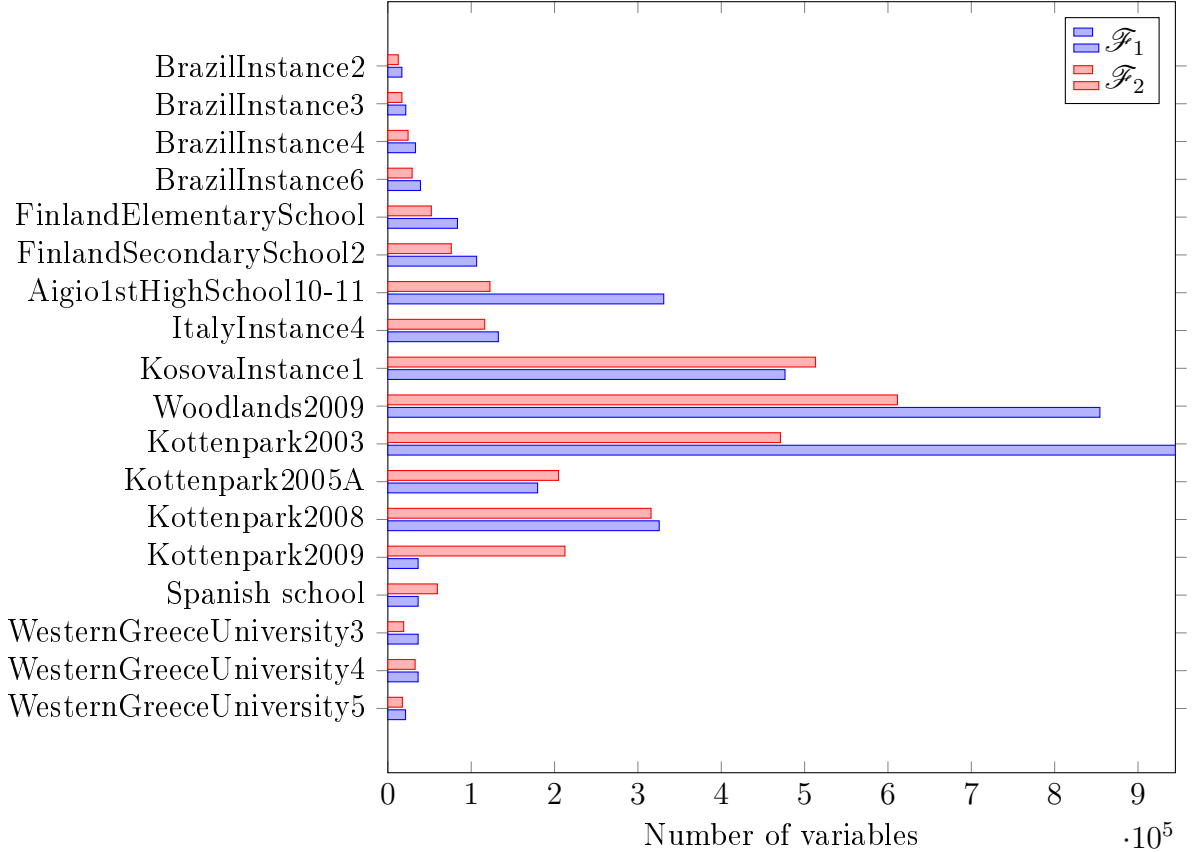
From the analysis of Table 2, it can be concluded that NBT and LXY were the most effective cuts, improving the linear relaxation of all considered instances. NBT affects variables $q_{r,t}$, which are essential to calculate resource related constraint penalties. LXY plays a central role since variables $y_{se,t}$ define the time assignment of the events and it provides a stronger link within the main variable, $x_{se,t,er,r}$. MCF does not affect instances FinlandElementarySchool, Woodlands2009 and Spanish School. Indeed, MCF strengthens the definition of Limit Idle Times constraints and this constraint does not apply to these instances. LYQ is effective when the assignment and preference of resources applies to the instances. It was also effective to strengthen

the link events specification. As expected, CBT only affects the instances whose this constraint applies to. Finally, it is possible to note that the characteristics of the instances provide valuable information for selecting the cuts to be active. Therefore the activation of cuts could be tailored according to the features of the input problem.

4.3. Model Dimensions

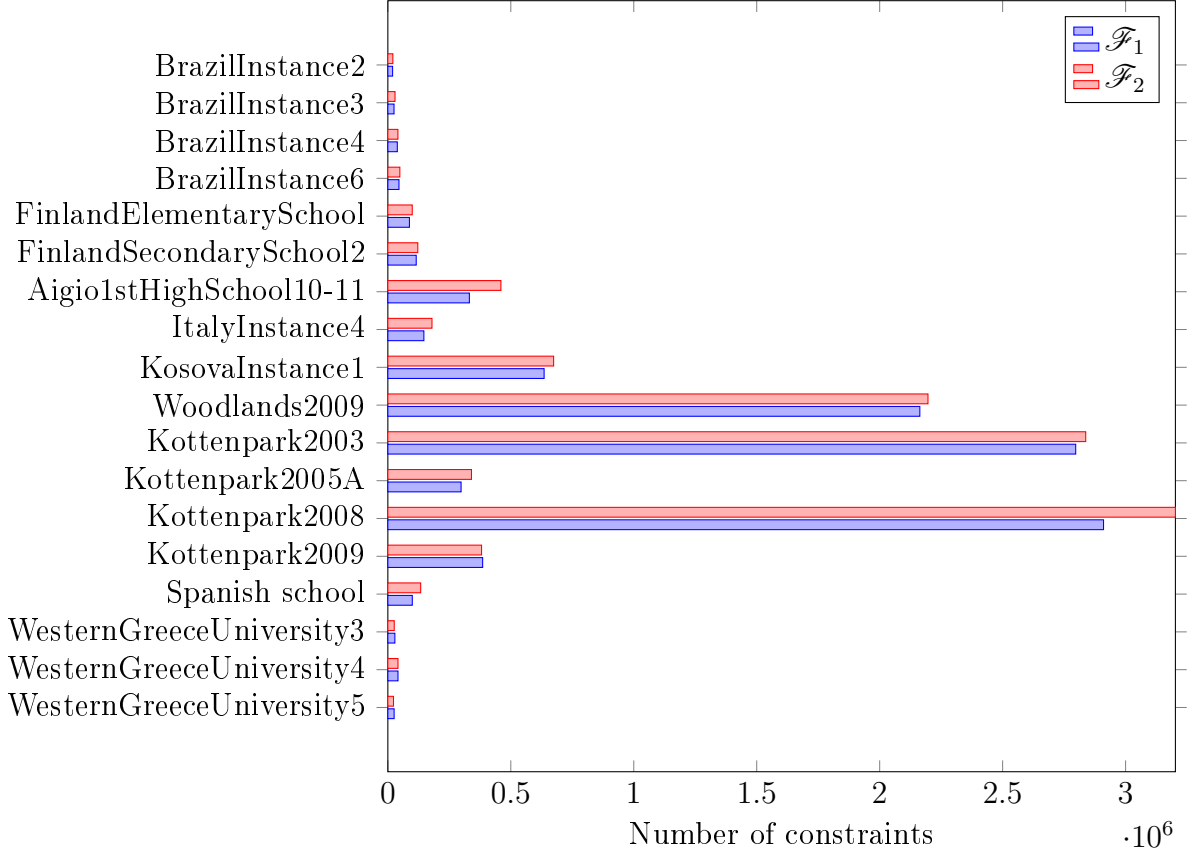
Figures 4, 5 and 6 represents, respectively the number of variables, constraints and non-zeros of formulations \mathcal{F}_1 and \mathcal{F}_2 applied to the instances of XHSTT-2011-hidden archive.

Figure 4: Comparison of the number of variables of \mathcal{F}_1 and \mathcal{F}_2 for each instance in XHSTT 2011 hidden.



The number of variables in \mathcal{F}_2 is slightly higher than in \mathcal{F}_1 in most of the cases. This is mainly due to the new variables introduced to represent

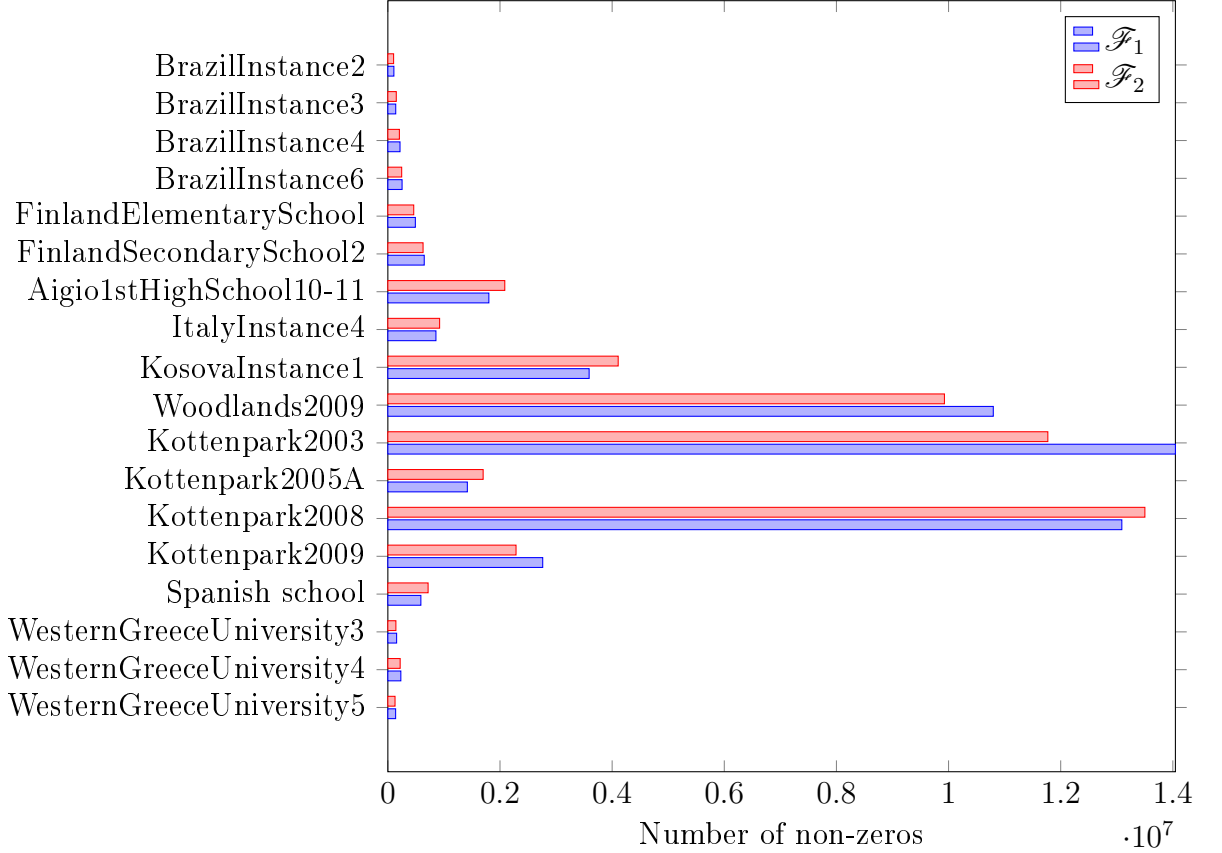
Figure 5: Comparison of the number of constraints of \mathcal{F}_1 and \mathcal{F}_2 for each instance in XHSTT 2011 hidden.



some of the arcs in the multicommodity flow reformulation. For some instances the increase rate is higher because they have a high number of resources. It forces the model to create a lot of graphs and therefore new variables. An odd behaviour can be observed for KosovaInstance1, whereas the number of variables reduces in \mathcal{F}_2 . This happens because a large number of variables to represent sub-events of infeasible duration is not generated in \mathcal{F}_2 .

Although the cuts demand new constraints to the models, the number of constraints is slightly smaller in \mathcal{F}_2 compared to \mathcal{F}_1 for most cases. It happens because of the reformulations, whereas several constraints can be dropped from the \mathcal{F}_1 . One example is Inequality (38), that generates a large number of constraints but can be removed when considering the multicommodity flow reformulation in \mathcal{F}_2 .

Figure 6: Comparison of the number of non-zeros of \mathcal{F}_1 and \mathcal{F}_2 for each instance in XHSTT 2011 hidden.



The number of non-zeros depends heavily on problem features. Therefore it is sometimes higher in \mathcal{F}_1 and sometimes higher in \mathcal{F}_2 as it can be seen in Figure 6.

4.4. Linear Relaxations

Table 3 presents linear relaxation of both original (\mathcal{F}_1) and alternative (\mathcal{F}_2) formulations over the instances of XHSTT-2011 and their running times as well. Column \downarrow Gap presents the percentage of decrease of the gap from using (\mathcal{F}_1) to using (\mathcal{F}_2). The table also presents the best known solution for these instances (\mathcal{UB}). When marked with a star, the best known solution is optimal. Instances whose optimal solution cost is zero were not included.

From Table 3, it can be noticed that the linear relaxations provided by \mathcal{F}_2

Table 3: Comparison between linear relaxations of \mathcal{F}_1 and \mathcal{F}_2

Instance	UB	Time (s)		\mathcal{LB}		\downarrow Gap
		\mathcal{F}_1	\mathcal{F}_2	\mathcal{F}_1	\mathcal{F}_2	
BrazilInstance2	5*	0.2	0.4	0.5	4.0	70%
BrazilInstance3	24*	0.2	0.4	0.0	2.0	8%
BrazilInstance4	51*	0.5	1.1	8.0	32.5	48%
BrazilInstance6	35*	2.5	4.4	2.0	16.0	40%
FinlandElementarySchool	3*	6.1	10.9	0.0	0.0	0%
ItalyInstance4	27*	15.8	25.6	0.0	15.0	56%
Kottenpark2003	420	5h+	5h+	0.0	0.0	0%
Kottenpark2005	784	5h+	5h+	0.0	86.0	11%
Kottenpark2008	15463	5h+	5h+	2.3	2904.0	19%
Kottenpark2009	5095	3115.6	8751.9	0.1	179.0	4%
Spanish school	335	24.4	23.4	54.9	305.0	75%
WesternGreeceUniversity3	5*	1.8	2.3	2.0	5.0	60%
WesternGreeceUniversity4	3*	6.5	3.4	0.0	0.0	0%

are significantly stronger than the ones from \mathcal{F}_1 , leading to an average of 32% of gap reduction. The processing times were slightly longer in \mathcal{F}_2 in most cases. For Kottenpark2009 in special, the new formulation was considerably more time consuming ($2.8\times$). However it led to a huge linear relaxation improvement. Such a lower bound was even better than the previously best known for this instance in the integer model (see Table 5). The lower bound of 2904 for Kottenpark2008 was also better than the previously best known bound.

4.5. Integer Model

Table 4 presents the integer results for both original and alternative formulations under one hour of time limit. Instances in which neither of the formulations achieved a feasible solution within the given time limit were omitted. Optimal bounds are marked with a star.

From the analysis of Table 4 it can be seen that \mathcal{F}_2 found the optimal lower bound for eight out of ten instances against five optimal lower bounds in \mathcal{F}_1 . Regarding the upper bounds the alternative formulation was also superior, finding five optimal bounds against three from the original formulation.

Table 4: Integer Programming results for formulations \mathcal{F}_1 and \mathcal{F}_2 within 1 hour time limit.

Instance	\mathcal{F}_1			\mathcal{F}_2		
	\mathcal{LB}	\mathcal{UB}	Gap	\mathcal{LB}	\mathcal{UB}	Gap
BrazilInstance2	4.5	10	0.55	5.0*	5*	0.00
BrazilInstance3	21.3	24	0.11	24.0*	24*	0.00
BrazilInstance4	49.7	138	0.64	51.0*	51*	0.00
BrazilInstance6	17.5	224	0.92	35.0*	217	0.84
FinlandElementarySchool	3.0*	3	0.00	2.7	4	0.33
Aigio1stHighSchool10-11	0.0*	0*	0.00	0.0*	0*	0.00
ItalyInstance4	27.0*	11,244	1.00	27.0*	15,348	1.00
WesternGreeceUniversity3	5.0*	9	0.44	5.0*	6	0.17
WesternGreeceUniversity4	0.0	24	1.00	2.0	8	0.75
WesternGreeceUniversity5	0.0*	0*	0.00	0.0*	0*	0.00

Although some progress towards finding better bounds for timetabling was done in this work, neither of the formulations were able to find any feasible solution for the eight remaining instances in XHSTT-2011-hidden. This result shows that this problem is still challenging for the research community, since exact methods are still not able to handle a considerable set of instances of this problem.

4.6. Improving Best Known Bounds

The automated timetabling research community keeps a track of the best known bounds for timetabling instances over XHSTT-2014 archive⁴. The aim of these experiments is to find new lower bounds or best known solutions (upper bounds) for this archive. To compute new lower bounds, the full integer programming model of alternative formulation \mathcal{F}_2 was executed. To improve the best known solutions (upper bounds) the alternative formulation was considered instead of the original formulation in a Matheuristic framework [15]. Instances whose optimal solution was known and proven were not considered. These experiments were performed during one month in a multi-core cluster, allowing each process to use four threads.

The new formulation was able to find and to prove the optimality for two new instances from XHSTT-2014, namely AU-SA-96 and AU-TE-99. These

⁴<https://www.utwente.nl/ctit/hstt/archives/XHSTT-2014/>

Table 5: New best known lower (\mathcal{LB}) and upper (\mathcal{UB}) bounds obtained using the new formulation \mathcal{F}_2

Instance	\mathcal{LB}	\mathcal{UB}	New \mathcal{LB}	New \mathcal{UB}	$\uparrow \mathcal{LB}\%$	$\downarrow \mathcal{UB}\%$
AU-BG-98	0	415	0	129	0%	69%
AU-SA-96	0	2	0	0*	0%	100%
AU-TE-99	0	33	20	20*	100%	100%
DK-FG-12	285	1514	412	1300	31%	14%
DK-HG-12	(7, 0)	(12, 2611)	(7, 0)	(12, 2356)	0%	10%
DK-VG-09	(0, 0)	(2, 2718)	(2, 0)	(2, 2329)	100%	14%
ES-SS-08	334	335	334	335	0%	0%
NL-KP-03	0	420	0	199	0%	53%
NL-KP-05	89	784	89	433	0%	45%
NL-KP-09	170	6265	180	1620	6%	74%
UK-SP-06	0	(15, 1892)	0	(5, 4014)	0%	67%
US-WS-09	0	111	0	103	0%	7%

instances are challenging and have been open for several years. In most of these instances events are constrained by Link Events. The reformulation of this constraint was very effective on these cases. AU-SA-98, from the same instance set (Australian instances), is still open but had its upper bound diminished in 69%. This reformulation of Link Events was also very effective on the Dutch instances (NL-KP-03, NL-KP-05 and NL-KP-09) whose upper bounds drastically fell as well.

Besides the achievement of several new upper bounds, new lower bounds were also produced for AU-TE-99, DK-FG-12, DK-VG-09 and NL-KP-09. Surprisingly, the little gap in ES-SS-03 could not be closed. Even though this instance is not large when compared to the others, it is still challenging for the automated timetabling research community.

5. Concluding Remarks

This paper presented new integer programming techniques for the Educational Timetabling problem. The developed cuts and reformulations drastically increased the lower bound of the linear relaxation for most of the instances in the XHSTT-2011-hidden archive. In exchange, a small increase on the running times for the linear relaxation was observed. The computational experiments also shown that the resulting integer programming models

from the proposed formulation are more effectively solved for most of the instances. The new formulation was also successful to find new bounds for the XHSTT-2014 archive: four new lower bounds and eleven new upper bounds were found. The optimality was found and proven for two of these instances.

Future work will focus on developing a column generation approach for this problem and embedding it in a branch-and-cut algorithm. The search for additional cuts based on the features of each instance is also subject of future research.

Acknowledgment

The authors would like to thank the Brazilian agencies CAPES, CNPq, and FAPEMIG for the financial support.

References

References

- [1] S. Even, A. Itai, A. Shamir, On the complexity of timetable and multicommodity flow problems, *SIAM Journal of Computing* 5 (4) (1976) 691–703.
- [2] C. C. Gotlieb, The construction of class-teacher time-tables, in: *Proc. IFIP Congress, Munich*, North Holland Pub. Co., Amsterdam, 1963, pp. 73–77.
- [3] T. Muller, ITC2007 solver description: a hybrid approach., *Annals OR* 172 (1) (2009) 429–446.
- [4] S. Ceschia, L. Di Gaspero, A. Schaerf, Design, engineering, and experimental analysis of a simulated annealing approach to the post-enrolment course timetabling problem, *Computers & Operations Research* 39 (7) (2012) 1615–1624.
- [5] A. V. Moura, R. A. Scaraficci, A GRASP strategy for a more constrained School Timetabling Problem, *International Journal of Operational Research* 7 (2) (2010) 152–170.

- [6] M. J. F. Souza, N. Maculan, L. S. Ochi, A grasp-tabu search algorithm for solving school timetabling problems, in: Maurício G. C. Resende; Jorge Pinho de Sousa. (Org.). METAHEURISTICS: Computer Decision-Making. Dordrech: Kluwer Academic Publishers, 2004, pp. 86 : 659–672.
- [7] L. Demir, S. Tunalı, D. T. Eliiyi, An adaptive tabu search approach for buffer allocation problem in unreliable non-homogenous production lines, *Computers & Operations Research* 39 (7) (2012) 1477–1486.
- [8] E. Burke, A. Eckersley, B. McCollum, S. Petrovic, R. Qu, Hybrid variable neighbourhood approaches to university exam timetabling, *European Journal of Operational Research* 206 (1) (2010) 46 – 53.
- [9] K. Nguyen, Q. Nguyen, H. Tran, P. Nguyen, N. Tran, Variable neighborhood search for a real-world curriculum-based university timetabling problem, in: 2011 Third International Conference on Knowledge and Systems Engineering, IEEE, 2011, pp. 157–162.
- [10] L. Di Gaspero, A. Schaerf, Multi-neighbourhood local search with application to course timetabling, in: *Practice and theory of automated timetabling IV*, Springer, 2003, pp. 262–275.
- [11] R. P. Badoni, D. Gupta, A hybrid algorithm for university course timetabling problem, *Innovative Systems Design and Engineering* 6 (2) (2015) 60–66.
- [12] E. K. Burke, J. P. Newall, A multistage evolutionary algorithm for the timetable problem, *Evolutionary Computation, IEEE Transactions on* 3 (1) (1999) 63–74.
- [13] M. M. El-Sherbiny, R. A. Zeineldin, A. M. El-Dhshan, Genetic algorithm for solving course timetable problems, *International Journal of Computer Applications* 124 (10).
- [14] H. G. Santos, T. A. M. Toffolo, R. A. M. Gomes, S. Ribas, Integer programming techniques for the nurse rostering problem, *Annals of Operations Research* 239 (1) (2016) 225–251.
- [15] G. H. Fonseca, H. G. Santos, E. G. Carrano, Integrating matheuristics and metaheuristics for timetabling, *Computers & Operations Research* 74 (2016) 108 – 117.

- [16] C. Barnhart, E. L. Johnson, G. L. Nemhauser, M. W. Savelsbergh, P. H. Vance, Branch-and-price: Column generation for solving huge integer programs, *Operations research* 46 (3) (1998) 316–329.
- [17] P. Avella, I. Vasilev, A computational study of a cutting plane algorithm for university course timetabling, *Journal of Scheduling* 8 (6) (2005) 497–514.
- [18] H. G. Santos, E. Uchoa, L. S. Ochi, N. Maculan, Strong bounds with cut and column generation for class-teacher timetabling, *Annals OR* 194 (1) (2012) 399–412.
- [19] A. J. Conejo, E. Castillo, R. Minguez, R. Garcia-Bertrand, *Decomposition techniques in mathematical programming: engineering and science applications*, Springer Science & Business Media, 2006.
- [20] G. Post, J. Kingston, S. Ahmadi, S. Daskalaki, C. Gogos, J. Kyngas, C. Nurmi, N. Musliu, N. Pillay, H. Santos, A. Schaerf, XHSTT: an XML archive for high school timetabling problems in different countries, *Annals of Operations Research* 218 (1) (2014) 295–301.
- [21] G. Post, L. Di Gaspero, J. Kingston, B. McCollum, A. Schaerf, The third international timetabling competition, *Annals of Operations Research* (2013) 1–7.
- [22] G. H. Fonseca, H. G. Santos, E. G. Carrano, T. J. Stidsen, Modelling and Solving University Course Timetabling Problems Through XHSTT, in: *PATAT '16 Proceedings of the 11th International Conference on the Practice and Theory of Automated Timetabling*, 2016, pp. 127–138.
- [23] S. Kristiansen, M. Sørensen, T. Stidsen, Integer programming for the generalized high school timetabling problem, *Journal of Scheduling* (2014) 1–16.
- [24] S. Brizo, H. Santos, Automatic integer programming reformulation using variable neighborhood search, in: *Proceedings of the 4th International Conference on Variable Neighborhood Search*, 2016.
- [25] M. Fischetti, A. Lodi, Optimizing over the first chvátal closure, *Mathematical Programming* 110 (1) (2007) 3–20.

- [26] A. P. Dorneles, O. C. de Araújo, L. S. Buriol, A column generation approach to high school timetabling modeled as a multicommodity flow problem, *European Journal of Operational Research* 256 (3) (2017) 685 – 695.